

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
26 July 2001 (26.07.2001)

PCT

(10) International Publication Number  
**WO 01/53898 A1**

(51) International Patent Classification<sup>7</sup>: **G05B 13/02**

(21) International Application Number: **PCT/US00/30980**

(22) International Filing Date:  
9 November 2000 (09.11.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
60/164,220 9 November 1999 (09.11.1999) US

(71) Applicant: **ANTRIM DESIGN SYSTEMS, INC.**  
[US/US]; 5550 Scotts Valley Drive, Suite 300, Scotts  
Valley, CA 95066 (US).

(72) Inventors: **ELLIS, Geoffrey**; 212 Germaine Avenue,  
Santa Cruz, CA 96065 (US). **LIM, Stephen**; 195 Oak

Creek Boulevard, Scotts Valley, CA 95066 (US). **DEM-  
LER, Michael, J.**; 153 Zinfandel Circle, Scotts Valley,  
CA 95066 (US).

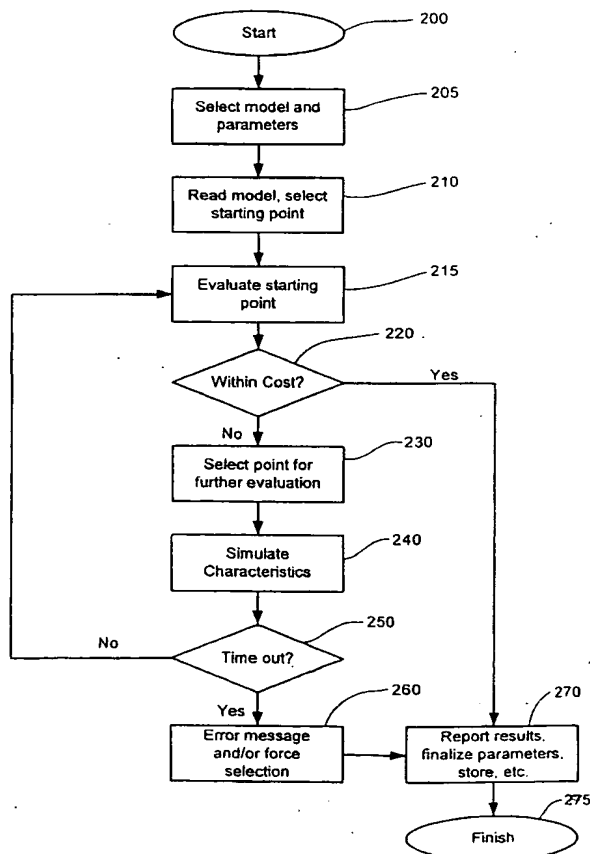
(74) Agents: **MEYER, Sheldon, R.** et al.; Fliesler, Dubb,  
Meyer and Lovejoy LLP, Suite 400, Four Embarcadero  
Center, San Francisco, CA 94111-4156 (US).

(81) Designated States (*national*): AE, AG, AL, AM, AT, AU,  
AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ,  
DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR,  
HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR,  
LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ,  
NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM,  
TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM,  
KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian  
patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European

[Continued on next page]

(54) Title: **LEARNING MODE OPTIMIZATION**



(57) Abstract: Given a database of performance values-point associations that grows every time optimization is run on that same circuit (but with different performance specifications), the optimizer gets progressively more efficient by evaluating a starting point (215) and evaluating a within cost question (215). If the cost is within a parameter chosen the results are stored (270). If the cost is not within (215) the parameters then an iteration is performed (220-250). It is more efficient because the probability of beginning optimization with a good starting point close to the optimum solution also increases with increased information in the database. This invention therefore enhances the speed of the class of optimizers that uses search-based algorithm.



patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

**Published:**

— *with international search report*

- 1 -

**LEARNING MODE OPTIMIZATION**

5

**COPYRIGHT NOTICE**

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

15

This application claims priority from provisional application "Learning Mode Optimization", Application No. 60/164,220, filed November 9, 1999, and specifically incorporated herein by reference.

20

**Field of the Invention:**

The invention relates generally to systems used for the simulation and synthesis of electronic circuits, and specifically to a system and a method for optionally synthesizing a circuit based on a learned knowledge of circuit criteria.

25

**Background of the Invention:**

In optimizing a circuit to meet a set of performance specifications, independent design parameters such as the physical dimensions of individual devices in the circuit (lengths and widths of transistors,

- 2 -

resistances and capacitances) are varied. When the circuit is large, that is, the number of devices in the circuit is high, the search space that the optimizer goes through can be prohibitively vast. Consider the following example:

5           A circuit has 10 design parameters that matter (i.e. they affect the performance characteristics of interest). For simplicity let us consider only lengths and widths of transistors. That is, the circuit has 5 transistors. Consider that each parameter can be varied between 1 $\mu$ m and 10 $\mu$ m, and the minimum step size is 0.1 $\mu$ m. That is, the optimizer is not allowed  
10          to vary a parameter by less than 0.1 $\mu$ m. The number of possible configurations for the circuit is:

$$(10 * 10 + 1) ^ 10$$

15          that is 1.104622e+20 (110,462,200,000,000,000). We may call each of these a "point" in the solution space. If each point takes just 1ns for the optimizer to evaluate, the worst case scenario of visiting all points would take 3502.7 years. That is, even visiting a 1,000th of that solution space would take 3.5 years, which is clearly unacceptable. Clearly, an  
20          efficient way is needed to drastically prune down the number of points necessary for the optimizer to visit during its search.

As described above, the set of all possible solutions to a multivariate problem can cover a vast space. Searching such a vast space of circuit design parameters in a random fashion results in an inefficient,  
25          time consuming, and possibly endless set of evaluations. In one prior art reference "Maelstrom: Efficient Simulation-Based Synthesis of Custom Analog Cells", Michael Krasnicki, Rodney Phelps, Rob A. Rutenbar, L. Richard Carley (Carnegie-Mellon University), Design Automation

- 3 -

Conference, June, 1999, herein incorporated by reference as "*Krasnicki, et al.*", an approach that involves using massively parallel computing power to enhance the speed of search is proposed. Such an approach requires the expense of not only the computing resources but also the software licenses required for each computer to run the circuit simulation in order to measure the required circuit performance.

In practice, in order to reduce the search time to an acceptable duration, an optimization algorithm will usually require a starting point that reduces this vast design parameter space to a neighborhood of proposed solutions in close proximity to other known working solutions.

In another prior art reference, HSPICE: HSPICE User's Manual, MetaSoftware Inc, (now Avant! Corporation), herein incorporated by reference as "HSPICE", a user of the optimizer must provide that starting point based on his knowledge of the circuit. Also, in the prior art, subsequent optimizations of the same circuit cause the optimizer to start over again from the beginning without benefitting from the knowledge of past evaluations.

#### **Summary of the Invention:**

In the new method for learning mode optimization which we disclose herein, the optimization algorithm has the ability to automatically choose a starting point by comparing the objective circuit performance specification to results of prior evaluations stored in a database. This increases efficiency by eliminating repetitive evaluations of known working solutions. In addition, as new working solutions are evaluated, they are added to the database for future reuse.

In one embodiment the invention comprises a method of building a device according to a reference specification comprising the steps of:

- 4 -

identifying a performance specification for a device; searching a database of performance characteristic sets each linked to a set of parameters for building said device; selecting a set of parameters linked to a performance characteristic set closely matching said performance specification; and  
5 utilizing the selected set of parameters in an optimization routine to find an optimal set of parameters for building said device.

In another embodiment the invention comprises a method for determining design parameters for an electronic circuit synthesis, comprising the steps of: inputting a desired characteristic of an element of  
10 an electronic circuit; accessing a database of entries wherein each entry in the database links a design parameter of a circuit element to a performance characteristic of said circuit element; and, determining a design parameter for a circuit element such that the performance characteristic of said circuit element optimally matches the desired characteristic for the element.

15 In yet another embodiment the invention comprises A method for determining design parameters for an electronic circuit synthesis, comprising the steps of: inputting a synthesis plan wherein said synthesis plan comprises a series of desired characteristics together with target values and value limits for each element of an electronic circuit; using the  
20 synthesis plan as a lookup into a database wherein each entry in the database links a design parameter of a circuit element to a performance characteristic of said circuit element to find a design parameter in the database that is associated with the synthesis plan; and, determining a design parameter for a circuit element such that the performance  
25 characteristic of said circuit element optimally matches the desired characteristic for the element, by evaluating the entry to see if it satisfies the desired characteristic, calculating a cost associated with the entry, and comparing the cost associated with a previously evaluated entry, and if it

- 5 -

does not satisfy the desired characteristics then locating a new entry in the database and reevaluating the entry to see if it satisfies the desired characteristic, repeating the steps of evaluating and locating entries until the entry satisfies the desired characteristic, and, reporting the design  
5 parameter associated with the entry that satisfies the desired characteristic.

**Brief Description of the Drawings:**

Figure 1 is an illustration of a minimize optimization method in  
10 accordance with an embodiment of the invention.

Figure 2 is an illustration of a maximize optimization method in accordance with an embodiment of the invention.

Figure 3 is an illustration of a goal optimization method in accordance with an embodiment of the invention.

15 Figure 4 is an illustration of an output of cost from the optimization in accordance with an embodiment of the invention.

Figure 5 is an illustration of a cost graph produced in accordance with an embodiment of the invention.

20 Figure 6 is an illustration of a sample lookup database file in accordance with an embodiment of the invention.

Figure 7 is an illustration of an example of an optimization process in accordance with an embodiment of the invention.

Figure 8 is a flowchart of an optimization method in accordance with an embodiment of the invention.

25

**Description of the Preferred Embodiment:**

The following glossary is provided to assist the reader in understanding some key terms used throughout this document.

- 6 -

Cost: A metric that gives an indication of the quality of the solution. Each performance value is measured and checked against the corresponding performance specification, and its 'closeness' to the performance specification determines how much it contributes to the cost. If all  
5 performance specifications are met, the cost is zero.

Design parameter: A circuit variable that the optimizer varies, such as transistor length and width. Each design parameter has a valid range the values within which the optimizer is allowed to use.  
10

Lookup model: A lookup model, if it exists, has one or more points.

Performance characteristic: A variable that characterizes a particular performance feature of the circuit, such as power consumption, slew rate,  
15 gain.

Point: A set of design parameters, and corresponding performance characteristics for a given circuit characterized in a given process.

20 Starting point: The first point used by the optimizer to begin its search.

Performance specification: A constraint that the user specifies on a performance characteristic. The optimizer tries to find a point that meets each of the performance specifications. A performance specification may  
25 have a single limit (upper limit or lower limit on a performance characteristic) or both limits.



- 7 -

Performance value: The measured value for a particular performance characteristic given a point.

5 Solution space: The set of all possible points given the set of design parameters and their value ranges.

Solution: A set of design parameter values determined by the optimizer.

10 Optimum solution: A solution whose performance values all meet the user's performance specifications and therefore has zero cost.

#### Objectives and Cost Functions

15 Optimization is the process of minimizing or maximizing a mathematical function. The function to be optimized is often called an objective function. In a circuit design optimization problem, an objective function describes the relationship between design parameters and a performance characteristic. For example, a single objective function could define a filter's frequency response as a function of the R, L, and C values of the network components. Here, optimizing the function adjusts  
20 component values to meet a desired response.

25 In many circuit design problems the objective functions are not known or cannot be easily derived analytically. An optimization methodology based solely on the formulation of an analytic model is inadequate. Practical circuit optimization requires algorithms to develop objective functions indirectly by evaluating models whose behavior represent the characteristic to be optimized. An electrical network constructed of transistor models implicitly describes electrical behavior of a circuit, which can be extracted by simulation in SPICE without requiring

- 8 -

a designer to provide an analytical expression. Consequently, many circuit optimization techniques have been based on analysis of circuit simulations. The techniques provided by an embodiment of the invention allow optimization to be performed most efficiently on behavioral models of circuit characteristics. Behavioral models can be combined with SPICE models to form a hierarchical synthesis plan.

In most circuit designs, several performance specifications must be met simultaneously, which in turn requires that a problem be formulated so that parallel optimization of multiple objectives can be performed. The invention uses synthesis plans to easily represent such performance specifications. A synthesis may contain one or more performance specifications. Each performance specification in a synthesis plan describes one objective for optimization. The invention optimizes multiple objectives in parallel by simultaneously evaluating test benches that contain the required synthesis models.

#### Cost Function

In circuit design, optimization objectives are generally dependent functions of the design variables. A method is required for formulating the optimization problem so trade-offs can be evaluated between possibly competing objectives. To optimize multiple objectives in parallel, the invention uses individual performance specifications in a synthesis plan to formulate a cost function. The cost function converts a vector of objectives to a single expression based on the weighted sum of the individual objectives.

A cost function is a form of objective function based on the differences between the measured performance results from evaluations of synthesis models and the objectives defined in the performance

specifications. Cost is a dimensionless, non-negative number, allowing results of performance measurements with different dimensions (e.g. volts, Hz, dBs) to be evaluated together. Each performance measurement is used to develop one term of the total cost equation. In one embodiment in which

5 the invention is incorporated with a computer program performing the steps of the invention, the total cost function is constructed by scaling each individual cost term proportional to the weights specified in a set of initial performance specification or `set_perf_spec()` commands. A high cost indicates an undesirable solution. Optimization algorithms used in the

10 invention seek to minimize the cost function, i.e., achieve a value of zero total cost, so all of the performance specifications are met.

#### Use of Targets, Limits and Effort Levels to Control Optimization

The form of an individual cost calculation is established when a

15 designer specifies an optimization objective and provides values for the limit and target values in a performance specification or `set_perf_spec()` command. The form of the performance specification and the choice of effort used in an `optimize()` command (which starts the optimization process), determine the criteria applied for achieving a successful

20 optimization. When low effort is specified, the optimizer attempts to find a solution where each of the limits on performance characteristics specified by the designer is met. When such a solution is found, optimization terminates. Targets do not affect the termination of the optimizer. However, when high effort is specified, the optimizer searches for a solution which

25 meets each of the targets in addition to each of the limits.

- 10 -

### Minimize Optimization

Figure 1 shows a minimize optimization method in accordance with an embodiment of the invention. In a minimize optimization an upper limit and a target (which may be undefined) can be specified for the performance objective. This may be represented as:

set\_perf\_spec(" perf\_spec", "min", "-", target, ulim, weight);

- For any performance measurement less than the target value, cost is defined as zero. Cost increases linearly to a value of one at the specified upper limit (ulim).
- For a result beyond the upper limit, cost continues to increase linearly, proportional to the amount by which the upper limit is exceeded.

### Cost Function for a Minimize Optimization

- If only an upper limit (ulim) is specified in a minimize optimization, the contribution to cost is zero for any performance measurement below the limit. For a result exceeding the limit, cost increases linearly from one as the measured result increases.

### Maximize Optimization

Figure 2 shows a maximize optimization method in accordance with an embodiment of the invention. In a maximize optimization a lower limit and an optional target may be specified for the performance objective.

set\_perf\_spec(" perf\_spec", "max", llim, target, "-", weight);

- For any performance measurement greater than the target value, the cost is defined to be zero.
- For a measured result less than the target value, the cost increases

- 11 -

linearly to a value of one at the specified lower limit (llim).

- For a result below the lower limit cost increases linearly from one as the measured result decreases.

#### 5 Cost Function for Maximize Optimization

- If only a lower limit (llim) is specified in a maximize optimization, the contribution to cost is zero for any performance measurement above the limit. For a result that is less than the limit, the reported cost increases linearly toward a value of one as the difference between  
10 the limit and the measured result increases.

#### Goal Optimization

Figure 3 shows a goal optimization method in accordance with an embodiment of the invention. A goal optimization requires the specification  
15 of both upper and lower limits in the set\_perf\_spec() command:  
set\_perf\_spec("perf\_spec", "goal", llim, target, ulim, weight);

A goal optimization can be performed by specifying the target directly, or by using the default value at the midpoint between the lower and upper limits. In a goal optimization a cost of zero occurs only when an exact  
20 match is achieved between a performance measurement and the target specification.

- For a result above or below the target, the contribution to cost increases linearly toward a value of one at the upper and lower limit.
- When the result either exceeds the upper or is less than the lower  
25 limit, cost continues to increase linearly to values greater than one in proportion to the amount by which the result goes outside the limit.
- In high effort mode, optimization continues to attempt to lower cost toward zero until a limit has been reached for iterations or time.

- 12 -

- In low effort mode any result between the upper and lower limits is acceptable.

To perform a goal optimization, a designer may choose to use low effort with narrow limits, rather than high effort. In many cases attempting to achieve an exact performance target is not realistic. Specifying low effort with narrow limits allows the optimizer to terminate with a satisfactory result, rather than continue to iterate when very low cost results have already been achieved.

#### Cost Display

During execution of a synthesis plan, one embodiment of the invention outputs a textual display of progress toward minimizing the cost function. The data displayed for minimum cost at each iteration is the measurement of the lowest cost achieved so far. An example of such a cost display is shown in Figure 4. Cost displayed may stay at the same value for several iterations while the optimizer explores points that result in higher cost. The 20th iteration in Figure 4 shows the lowest cost recorded.

#### Optimization Algorithms

There are many different algorithms for general purpose optimization. To adapt an algorithm for use in synthesis of analog integrated circuits, modifications must be made for the discrete nature of most design parameters, and to address multiple objectives and hard constraints. Optimization algorithms can generally be divided into those that analyze derivatives of the objective functions, and those that do not.

Algorithms that are not based on evaluation of derivatives perform a search for solutions by evaluation of models. A search may be random or

- 13 -

directed. In accordance with an embodiment of the invention the exploration algorithm adapts the Hooke-Jeeves method (described in detail in Kelley, C.T., "Iterative Methods for Optimization", Philadelphia, PA: Society of Industrial and Applied Mathematics (SIAM), Philadelphia 1999, and  
5 incorporated herein by reference as "Kelley") to perform analog circuit optimization through a directed search. Optimization algorithms that measure derivatives use information about the slope of a function as an indication of the direction in which a minimum may be found. These gradient algorithms can be more efficient than search methods when the  
10 functions being evaluated have no discontinuities in their first derivatives. In the invention, Newton's method, also described in Kelley, is used for gradient optimization. Since the calculation of derivatives at each evaluation of a synthesis model can require a significant number of computations, the invention uses an approximation method known as the BFGS formula  
15 (described in Broyden, Fletcher, Goldfarb, and Shanno, Broyden, C.G., "The Convergence of a Class of Double-Rank Minimization Algorithms", J. Inst. Maths. Applics., Vol. 6, pp. 76-90, 1970 incorporated herein by reference as "Broyden, et al.").

## 20 Hooke-Jeeves Exploration

In accordance with an embodiment of the invention, the Hooke-Jeeves technique for optimizing continuous functions has been adapted for use in circuit synthesis, where design parameters typically take discrete values according to the grid size of the fabrication process. The  
25 term grid size is well known to one skilled in the art. In addition to using a finite step size for each design parameter, exploration is also limited to the range specified by the upper and lower limits in a `set_opt_param()`

- 14 -

command. The set\_opt\_param() command is used to designate design parameters as optimization variables.

#### Initial min\_step Size

5           Using the Hooke-Jeeves exploration technique for each design parameter, the initial step size is approximately 10 percent of the difference between the maximum and minimum values specified on the set\_opt\_param() command. This percentage is approximate because the step size is always an integer multiple of the min\_step size (the minimum  
10   optimization step size) specified on the set\_opt\_param() command, i.e., all stepping is on grid. The term on grid is well known to one skilled in the art. In the exploration mode, the optimizer feature of the invention uses information provided in each set\_opt\_param() command to control the process of searching for a solution that meets all performance  
15   specifications.

          The first step always evaluates the synthesis model with the starting point values for the design parameters. If the performance objectives have not all been met by the starting point values, the exploration optimizer begins its search.

20

#### Cost Function Evaluation

          Exploration evaluates the cost function after each parameter step is taken to determine if progress has been made toward the performance objectives (i.e., cost has been reduced). If a step in one direction from the  
25   starting point does not result in progress, the opposite direction is tried.

          When progress is made in one direction, steps continue in that direction as long as cost continues to decrease or a limit in a parameter



- 15 -

value is reached, as seen in Figure 5 which illustrates a 40-iteration optimization graph of the cost, \$Cost.

The order of stepping parameters is executed in the reverse order of the set\_opt\_param() command statements during the initial exploration.

5 Because of this dependency in exploration, the number of iterations required to achieve an acceptable cost varies with the order in which parameters are evaluated. If one parameter is known to have a greater influence on the performance specifications than others, iterations may be saved if this parameter to set it as the first to be evaluated.

10 If the exploration algorithm finds a local minimum, but that minimum fails to meet the designer's performance requirements, the optimizer selects a new starting point and continues further exploration. Such exploration is always subject to the global limits of imax, the maximum number of iterations allowed, and tmax, the maximum time allowed, on the optimize() command which take precedence in terminating the exploration.

15 The optimizer uses the explore algorithm as its default if gradient is not specified in an optional argument of the optimize() command.

#### Newton's Method Gradient

20 Designating the gradient algorithm on the optimize() command employs Newton's enhanced method, a technique that derives the shape of an objective function by estimating its derivatives. Estimated second derivatives of a function determine the direction in which the optimizer searches for a minimum. From an initial starting point, this method proceeds  
25 toward some local minimum of the cost function. Newton's method has the potential to converge to a local minimum much more quickly than the Hooke-Jeeves method. However, it is not possible to know where the neighborhood of a local minimum might be. Moreover, due to numerical

- 16 -

noise generated in simulation and in measurement functions, actual results from the gradient may be slower than exploration for some designs.

#### Terminating Optimization

5 Optimization terminates before finding a local minimum if each of the designer's performance requirements is met, or if the -imax or -tmax limit is reached, whichever occurs first. If a local minimum is found, but that minimum fails to meet the designer's performance requirements, the optimizer selects a new starting point and performs a new local  
10 minimization in succeeding iterations.

As in the explore algorithm, if the gradient exploration finds a local minimum, but that minimum fails to meet the designer's performance requirements, the optimizer selects a new starting point and continues further exploration. Such exploration is always subject to the global limits  
15 of imax and tmax on the optimize() command which take precedence in terminating the exploration.

#### Learning Mode

Learn Mode is enabled by a -learn optional argument on the  
20 optimize() command. Specifying this option causes the optimizer results to be added to the lookup database. If no such database exists, it is created by an optimization in learn mode. A designer enables the Learning Mode with the -learn option during optimization to build a database of points that have been evaluated. For example, in one embodiment of the invention  
25 which provides a command line interface the designer may specify the -learn option as a text argument:

```
MSS 101> optimize("-learn", "-design", "vco_ckt", "-effort", "high", "-imax",  
20);
```

- 17 -

Saving the points in a database eliminates the need for further simulation of the same parameter values and speeds up subsequent optimizations, where a performance specification may match a value that was previously measured. A designer can also use the learning database to restart an optimization to refine the results by providing more iterations. When the optimization is restarted with Learning Mode enabled, the database is read before simulation is begun.

In Learning Mode, the optimization parameter values and resultant performance specification measurements are saved in the reference library.

In one embodiment, this data is written to a file in the view directory of the cell being optimized and is named lookup.db. An example of the contents of a lookup.db file are shown in Figure 6.

#### Starting Points and Lookup Database

Lookup Mode is enabled by a -lookup optional argument in the optimize() command. This causes the optimizer to use the previously stored results in the lookup database where possible, instead of performing a simulation. Also, in Lookup Mode, the optimizer uses as its starting point the lowest cost point in the lookup model. Learning Mode implies Lookup Mode.

Use of Learning Mode and Lookup Mode makes simulation results from previous optimizations of a given cell available in a new optimization and saves simulation time during optimization. Use of Learning Mode and Lookup Mode is a highly efficient way to use the optimizer. A lookup database is a set of entries, each containing a set of design parameter values and the corresponding performance characteristic values.

When Learning Mode or Lookup Mode is used, each performance characteristic in the cell should be listed in a set\_perf\_spec() command. A

- 18 -

performance characteristic not required for a given optimization can have its weight set to zero (0).

After a lookup database is updated in Learning Mode, the list of performance specifications in the set\_perf\_spec() commands should not be changed. If the set of performance specifications changes, simulation results in the existing lookup.db file may be invalid with respect to the new netlist and should be deleted before optimization begins. The lookup database can be stored in a file named lookup.db in the lib/cell/view directory.

- To use Learning Mode, the user must have read/write access to this file.
- To use Lookup Mode, only read access is required.

The system does not manage concurrent access to the lookup database.

#### Device Representation

In the present invention, a process of learning mode optimization allows that, as additional points are tried during synthesis, these points can be added to the lookup model, reducing the time required for some future synthesis using the given circuit, circuit cell, or cell. As a given circuit is optimized to various performance criteria, a database of learned data increases in size, making future optimizations run faster.

In accordance with one embodiment of the learning mode optimization feature of the invention, a database stores a set of points. The set of points may be in the form of a table of performance values versus point associations. Other data structures can be used for storing such points, including spreadsheets, relational databases, linked lists, neural nets and their equivalents. Also, the term 'point' can be used to refer to any or

- 19 -

more entries in such a data structure that represent a particular set of specifications. Given a set of performance specifications specified by the user, the optimizer looks up or references the table for the closest matching set of performance values and returns the associated point. If multiple  
5 points are available, the optimizer can select the best one as a starting point for optimization. The best start point is the one which would most quickly lead to a successful optimization.

The optimizer does not initially know which point actually is the best. To account for this, the optimizer selects the point with the lowest cost,  
10 based on the performance criteria specified by the user, as the starting point. This represents the optimizer's guess as to the best starting point. If the user-specified set of performance specifications exactly matches a set of performance values in the table, then the point associated with that set of performance values is the optimum solution and no further optimization  
15 is required. Otherwise, the optimizer uses that point as the starting point and performs an optimization.

The solution found by optimization, as well as all intermediate points visited during the optimization, are added to the database, keyed by the measured performance values for each of the points. In this way the  
20 database grows in knowledge, and, the optimizer has a means to learn from previous optimizations.

In general, as more points are stored in the lookup model, a better starting point can be selected. These points are typically found by a process of simulation, which can be expensive in terms of machine time.  
25 Therefore saving these points in the database saves future computational effort.

In most electronic designs, there are multiple design parameters and multiple performance characteristics. For example purposes, one may

- 20 -

consider the design of an electronic device (device1) having a frequency characteristic and a channel length design parameter. The electronic device has a lookup model with one point as shown in Table 1.

5

Point No.	Parameters	Characteristics
1	Channel Length 2.0 $\mu$ m	Frequency 5e6Hz

Table 1: Lookup Model Name: Device1

10        The lookup model in Table 1 indicates that Device1 has been simulated (or perhaps actually built and tested), using the parameter channel length at a value of 2.0  $\mu$ m. With this parameter, device1 has a characteristic frequency of 5e6Hz.

15        When learning mode optimization is invoked to determine parameters for a new device for example, one having a desired characteristic frequency of 1e7Hz, the model is read to determine a starting point. A starting point having characteristics closest to the desired characteristics (frequency of 1e7Hz) is selected. In this example, since only point 1 is present, it is selected as the starting point.

20        The optimizer then selects another set of parameters for further evaluation. Any selection process may be utilized to determine other parameter values for evaluation, including, for example, random number generation within upper and lower limits, step functions, preselected steps, gradient, etc. The evaluation process itself may use any simulation or  
25        measurement to determine the performance characteristics of the circuit.

Referring now to Figure 7, a graph 100 of frequency versus channel length is shown. The starting point, point 1, 110 is illustrated from the lookup model of Table 1. A second point, 120 illustrates a possible set of parameters for further evaluation, having increased channel length compared to the starting point. In this example, a frequency characteristic of the second point is simulated to be 1e6Hz.

Since the second point 120 represents a characteristic frequency in a same direction away (less than) from the desired frequency, and further from the desired frequency, as compared to the starting point, the optimizer then selects a channel length parameter less than the starting point (third point 130) and re-evaluates. Third point 130 is shown having a frequency of exactly or very closely matching the desired frequency. If the optimizer finds a point that is within the performance specifications set by the user, then a solution has been found and the optimization is completed.

Each of the points evaluated in finding the solution are then added to the lookup model. The updated lookup model is shown in Table 2.

Point No.	Parameters	Characteristics
1	Channel Length 2.0 $\mu$ m	Frequency 5e6Hz
2	Channel Length 2.5 $\mu$ m	Frequency 1e6Hz
3	Channel Length 1.5 $\mu$ m	Frequency 1e7Hz

Table 2: Lookup Model Name: Device1

- 22 -

Using the updated lookup model, future designs based on Device1 will now have three simulated points to select for an optimization starting point.

Many variations of the above described process for implementing learning mode are possible in light of the present disclosure. One implementation is illustrated in the flow chart of Figure 8, and includes features such as cost analysis 220, time out 250, error reporting 260, and results storage/reporting 270 (including update of the lookup table).

As shown in Figure 8, the process begins by selecting a model and associated parameters in step 205. A suitable starting point is selected in step 210. This point is evaluated in step 220 to determine whether it is within the cost restraints set by the system, user, or operator. If the point is within the prespecified cost the point is chosen and the results reported in step 270. If it is not within the specified cost then the point is added to a database in step 230, a new point is evaluated in step 240, and the process reiterates. A timeout step 250 causes the process to jump to an error step 260 should any step not take place within an appropriate timeframe.

## 20 Cost-based Lookup

The look up of points using performance values as the key is done as follows. The notion of cost is used. By definition, the optimum solution has zero cost. Deviation from the optimum has increasing cost, according to some cost function. With this definition, the user-specified set of performance specifications correspond to a solution that has zero cost, meaning, as described above, that if a point in the table is associated with a set of performance values that meet this set of performance specifications, then that point is considered the optimum solution. In



- 23 -

general, such a point may not exist. Using cost as a basis of comparison, the entries in the table of performance values-point associations are visited in some order. The cost of each set of performance values are evaluated. The point associated with the set of performance values that has a cost  
5 closest to zero or equal to zero, is returned.

#### Cell Revisions

Cell designs are likely to change over time. In one embodiment, the present invention includes capabilities to prevent attempts to optimize a  
10 circuit using a lookup model from a previous and incompatible revision of that cell. A cell, in addition to having a name, function, and library may also have a revision number. When a circuit is characterized, the learned data will include both the circuit name and the revision number.

In one embodiment, the revision number is of the form X.Y, where  
15 each of X and Y is an integer. Here X represents the major revision number, and Y the minor revision number. A change in X indicates a major change in the circuit design -- any learned data developed previously to the current circuit design is invalid. A change in Y indicates a minor revision -- previously developed learned data may be used, to find a starting point for  
20 optimization, however, the point needs to be validated by simulation. Even if such a data point appears to be optimal (i.e. zero cost), at least one simulation is performed.

A given process may also change over time. For similar reasons, circuit learned data will include both the process name and revision number,  
25 in the form X.Y as above.

In one embodiment, a circuit selection function is enhanced to accept both a circuit name and a revision number. Similarly, a process selection

- 24 -

function is also enhanced to accept both a process name and a revision number.

In principal, the circuit simulator may also have a major and minor revision number. Therefore, simulator revision numbers may also be  
5 tracked as part of the data management.

#### Distributed Databases

In practical application of the present invention, it is envisioned that, for a given customer organization, there will be a group responsible for  
10 installing and maintaining optimization software and circuit libraries. This group will characterize the circuit library in each of the processes that design engineers will use. This learned data will be provided to the entire organization. This group will also provide new cells and new process descriptions, and revisions to them. This information will be provided by the  
15 present invention on a read-only basis (through a network mounted file system, for example). Therefore other groups, such a design engineering group, would not be allowed to change this data. In one embodiment, the present invention includes the following repositories for learned data:

20 Work group level: a given work group can keep their shared data in a specified location. Only members of this work group may write the data. Members of other groups may read the data, if this group chooses to make it public (across the organization). Otherwise the data is completely private to the group. Access would be controlled through Operating system file  
25 permissions. If everyone in the organization shares the data, then the organization may comprise a single work group.

- 25 -

Individual engineer level: an engineer can keep data which is shared across various designs, in a specified location. Only this engineer may write the data. Others may read the data, if this engineer chooses to make it public across their work group, or to the entire organization. Access can be  
5 controlled through Operating system file permissions.

#### Data Sharing Commands

There can be explicit user commands to share or un-share repositories of learned data. Data may be imported read-only or read-  
10 write. Permissions may be set such that data cannot be imported read-write unless the owner of the data exported it read-write. When an optimization is started, the optimizer is given a list of learned data repositories to be read. The optimizer is also given a list of repositories to be written.

15

#### File Locking

File locking is utilized to avoid two separate programs writing to the same file at approximately the same time. That could result in loss of data or file corruption. Therefore, in one embodiment, an operating system file  
20 locking mechanism is used. No file learned data is read until a read lock is acquired. Similarly, the file is not written until a write lock is acquired. To avoid a deadlock, the following strategy may be utilized in handling locks:

- For reading, the optimizer locks only one file at a time. That file is read to completion, then unlocked.
- The optimizer will not wait forever to acquire a read lock on a learned data file. If, after a reasonable wait, the read lock cannot be acquired, the optimizer gives up on reading that file. Optimization

25

- 26 -

proceeds with using the data from any other learned data files that can be read. If no learned data file can be read, and if the user does not specify a start point, then optimization will fail.

- 5       –     For writing, the optimizer will lock only one file at a time. That file is updated, then unlocked.
- The optimizer will not wait forever to acquire a write lock on a learned data file. If, after a reasonable wait, a write lock cannot be
- 10       acquired, the optimizer will give up on writing that file.

#### File Selection

          Given a set of files available on a date to be read, the optimizer will select the best files to actually read (or write). The selection criteria will be

15       as follows:

- The circuit name and major rev number, as well as the process name and major rev number need to match those specified to the optimizer for sharing purposes.
- 20       –     Given a set of shareable files in a directory, only one of those files will be selected.
- The selected file is the one with both the matching circuit minor rev
- 25       and the matching process minor rev, if such a file exists. Otherwise, the file with the matching minor circuit rev and closest minor process rev is selected, if that file exists. Otherwise the file with the closest

- 27 -

minor circuit rev and closest minor process rev is selected. This selection mechanism is just for reading.

- 5        -        For writing, the file with the matching circuit minor rev and the matching process minor rev can be selected. A matching filename can be either an exact match, or a matching merged name, as described below. If no such file exists, a new file will be written to the specified directory.

10       Merging

As a maintenance function, it may become necessary to build a new learned data file from two or more existing files. The following rules apply to merges:

- 15       -        Only files for the same circuit name and major revision, and the same process name and major revision, may be merged.
- 20       -        Files with different minor revisions may be merged. The resulting file will have a compound name indicating the revision range(s) included. For example, when merging the files

vco\_1.2\_cmos7\_3.2

vco\_1.2\_cmos7\_3.3

vco\_1.4\_cmos7\_3.2

25

The resulting file name will be

vco\_1.2\_1.4\_cmos7\_3.2\_3.3

- 28 -

This file contains data for vco with versions in the inclusive range [1.2, 1.4] for cmos7 processes in the inclusive range [3.2, 3.3].

#### Listing of Files

5           A command may be provided to list the files that would be shared, and the type of sharing, for a specified library, circuit and process.

#### Deletion of Learned data

10           A command is provided to delete a learned data file. The deletion will not be performed if some other program has a read lock, or a write lock, on the file.

15           Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art. The invention may also be implemented by the preparation of application specific integrated circuits or by interconnecting an appropriate network of conventional component circuits, as will be readily apparent to those skilled in the art.

20           The present invention includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the processes of the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical  
25           cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

          Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the

- 29 -

general purpose/specialized computer or microprocessor, and for enabling the computer or microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, and user applications. Ultimately, such computer readable media further includes software for performing the present invention, as described above.

Included in the programming (software) of the general/specialized computer or microprocessor are software modules for implementing the teachings of the present invention, including, but not limited to, maintaining a database of parameters and related performance characteristics, simulating devices built according to specified parameters, altering parameters via mathematical iteration or other mechanisms, matching or selecting performance characteristics according to user specifications, and the display, storage, or communication of results according to the processes of the present invention.

Obviously, numerous modifications and variations of the present invention are possible in light of the above teachings. It is therefore to be understood that within the scope of the appended claims, the invention may be practiced otherwise than as specifically described herein.

20

- 30 -

**Claims:**

What is claimed is:

- 5        1.        A method, comprising the steps of:  
                 identifying a performance specification for a device;  
                 searching a database of performance characteristic sets each linked  
                 to a set of parameters for building said device;  
                 selecting a set of parameters linked to a performance characteristic  
10        set closely matching said performance specification; and,  
                 utilizing the selected set of parameters in an optimization routine to  
                 find an optimal set of parameters for building said device.
- 15        2.        The method according to Claim 1, wherein said step of utilizing  
                 comprises utilizing the selected set of parameters as a starting point in an  
                 optimization routine to find an optimal set of parameters for building said  
                 device.
- 20        3.        The method according to Claim 1, wherein said step of utilizing  
                 comprises the steps of:  
                 altering at least one parameter of the selected set of parameters (or  
                 the altered parameter set on a next repeat altering step) to produce an  
                 altered parameter set;  
                 determining performance of said device if built from the altered  
25        parameter set; and  
                 evaluating the performance of said device to determine a next  
                 alteration of at least one parameter of the altered parameter set to be  
                 altered in a next altering step; and,



- 31 -

repeating said steps of altering, determining, and evaluating until one of an optimal solution and closest to optimal solution is found.

4. The method according to Claim 3, wherein said step of determining  
5 comprises the step of simulating performance of said device constructed with said parameter set.
5. The method according to Claim 1, further comprising the step of:  
saving parameters and related performance characteristics of any  
10 intermediate solutions calculated by said optimization in said database.
6. The method according to Claim 1, wherein said device is one of an  
electronic and electrical circuit, said parameters are circuit parameters, and  
said performance characteristics are specifications for performance of said  
15 one of an electronic and electrical circuit.
7. The method according to Claim 6, wherein said circuit parameters  
are physical dimensions of individual devices in the circuit.
8. The method according to Claim 1, wherein said step of selecting,  
20 comprises:  
selecting a set of parameters linked to a performance characteristic  
set having a closest match to said first performance characteristic set.
9. A method for determining design parameters for an electronic circuit  
25 synthesis, comprising the steps of:  
inputting a desired performance specification of an element of an  
electronic circuit;

- 32 -

accessing a database of entries wherein each entry in the database links a design parameter of a circuit element to a performance characteristic of said circuit element; and,

5 determining a design parameter for a circuit element such that the performance characteristic of said circuit element optimally matches the desired performance specification for the element.

10 10. The method of Claim 9 wherein said step of inputting includes inputting a plurality of desired performance specifications of the element.

11. The method of Claim 9 wherein said step of inputting includes specifying value limits for the desired performance specification.

15 12. The method of Claim 9 wherein said step of inputting includes specifying a target value and value limits for the desired performance specification.

20 13. The method of Claim 9 wherein said step of inputting includes inputting a synthesis plan wherein said synthesis plan comprises a set of desired element performance specifications together with target values and value limits for each desired performance specification.

25 14. The method of Claim 9 wherein said step of accessing includes:  
using the desired performance specification as a lookup into the database; and,

finding the design parameter in the database that is associated with the lookup.

- 33 -

15. The method of Claim 14 wherein the database is in the form of a table of entries, and wherein each entry comprises a parameter value, and a characteristic values
- 5 16. The method of Claim 9 wherein said step of determining comprises:  
determining from the desired performance specification an initial entry in the database;  
evaluating the entry to see if the performance characteristic associated with that entry satisfies the desired performance specification;  
10 and,  
if it does not satisfy the desired performance specification then locating a new entry in the database and reevaluating the new entry to see if it satisfies the desired performance specification.
- 15 17. The method of Claim 16 further comprising the step of:  
repeating the steps of evaluating and locating entries until the entry satisfies the desired performance specification; and,  
reporting the design parameter associated with the entry that satisfies the desired performance specification.
- 20 18. The method of Claim 16 wherein said step of evaluating includes:  
calculating a cost associated with the entry; and,  
comparing the cost associated with a previously evaluated entry.
- 25 19. The method of Claim 16 wherein said step of evaluating includes:  
calculating a cost associated with the entry; and,  
comparing the cost with a pre-set cost limit set by the operator.

- 34 -

20. The method of Claim 16 wherein said step of locating a new entry includes:

creating a new entry if one does not exist.

5 21. The method of Claim 20 wherein the new entry is saved to the database for use in subsequent operation of the method.

22. The method of Claim 9 further comprising the step of:  
outputting the design parameters to an output device.

10

23. The method of Claim 22 further comprising the step of:  
synthesizing an electronic circuit in accordance with the design parameters.

15 24. A system for selecting design parameters of electronic devices for use in an electronic circuit, comprising:

an input device to allow input of a set of desired performance specifications;

20 a database having a set of entries, wherein each entry includes information about design parameters and associated performance characteristics of an electronic device;

an optimizer, for determining an entry in the database which optimally matches the set of desired performance specifications; and,

an output device for outputting the optimal design parameters.

25

25. The system of Claim 24 wherein the set of desired performance specifications include targets and limits associated with each specification.

- 35 -

26. The system of Claim 24 wherein the optimizer comprises:

a starting point determiner, which determines an initial entry as a starting point in the database;

5 a cost determiner, which calculates the cost of the starting point compared to the desired performance specification; and,

an entry creator, which creates new starting points in response to determining the cost of the previous starting point.

27. The system of Claim 24 further comprising:

10 a database updater, which updates the entries of the database with new entries that match the starting points created by the optimizer.

28. The system of Claim 27 wherein the database updater may be switched on or off by an operator of the method.

15

29. A method for determining design parameters for an electronic circuit synthesis, comprising the steps of:

inputting a synthesis plan wherein said synthesis plan comprises a desired performance specification of an element of an electronic circuit,  
20 together with target values and value limits for said performance specification;

using the synthesis plan as a lookup into a database wherein each entry in the database links a design parameter of a circuit element to a performance characteristic of said circuit element, to find a performance  
25 characteristic in the database associated with the synthesis plan; and,

determining a design parameter for a circuit element such that the performance characteristic of said circuit element optimally matches the desired performance specification for the element, by

- 36 -

evaluating the entry to see if it satisfies the desired performance specification,

calculating a cost associated with the entry, and

5 comparing the cost associated with a previously evaluated entry, and if it does not satisfy the desired performance specification then locating a new entry in the database and reevaluating the entry to see if it satisfies the desired performance specification,

repeating the steps of evaluating and locating entries until the entry satisfies the desired performance specification, and,

10 reporting the design parameter associated with the entry that satisfies the desired performance specification.

30. The method of Claim 29 wherein the database is in the form of a table of entries, and wherein each entry comprises an element parameter value and a characteristic value.

15

31. The method of Claim 29 further comprising the step of :  
creating a new entry if one does not exist; and,  
saving the new entry to the database for use in subsequent operation  
20 of the method.

32. The method of Claim 29 further comprising the step of:  
synthesizing an electronic circuit in accordance with the design parameters.

25

33. A computer readable medium having computer instructions thereon that when loaded into a computer causes the computer to perform the steps of:

- 37 -

accepting a desired performance specification of an element of an electronic circuit;

accessing a database of entries wherein each entry in the database links a design parameter of a circuit element to a performance characteristic of said circuit element; and,

determining a design parameter for a circuit element such that the performance characteristic of said circuit element optimally matches the desired performance specification for the element.

10 34. The computer readable medium of Claim 33 wherein said step of determining comprises:

determining from the desired performance specification an initial entry in the database;

15 evaluating the entry to see if it satisfies the desired performance specification; and,

if it does not satisfy the desired performance specification then locating a new entry in the database and reevaluating the entry to see if it satisfies the desired performance specification.

20 35. The method of Claim 34 wherein said step of locating a new entry includes:

creating a new entry if one does not exist.

25 36. The method of Claim 35 wherein the new entry is saved to the database for use in subsequent operation of the method.

37. The method of Claim 33 further comprising:  
outputting the design parameters to an output device.

- 38 -

38. The method of Claim 33 further comprising:  
synthesizing an electronic circuit in accordance with the design parameters.

5 39. A method of creating a database of circuit element design parameters for use in electronic circuit synthesis, comprising the steps of:  
inputting a desired performance specification of a circuit element;  
constructing a model set for the circuit element which links design parameters of said circuit element to the performance characteristics for  
10 said circuit element;  
determining, using the model set, a design parameter which satisfies the desired performance specification; and,  
storing the model set in a database as a learned model for use in subsequent circuit element design parameter determination.

15

40. The method of Claim 39 wherein said step of inputting includes inputting a plurality of desired performance specification of the circuit element.

20 41. The method of Claim 39 wherein said step of inputting includes specifying a value limit for the desired performance specification.

42. The method of Claim 41 wherein said step of inputting includes specifying a target value for the desired performance specification.

25

43. The method of Claim 39 wherein said step of inputting includes inputting a synthesis plan wherein said synthesis plan comprises a set of



- 39 -

desired circuit element performance specification together with target values and value limits for each of said performance specification.

5 44. The method of Claim 39 wherein said step of storing includes storing the learned model as a table entry in the database wherein the table entry comprises a design parameter value and an associated performance characteristic value.

10 45. The method of Claim 44 wherein said table entry includes a plurality of design parameter values and associated performance characteristic values.

15 46. The method of Claim 39 wherein said step of determining includes:  
referencing the model set to determine the performance characteristic associated with a design parameter specified by the model set; and,

comparing the performance characteristic, as determined by referencing the model set, with the desired performance specification.

20 47. The method of Claim 46 wherein said step of determining further comprises:

creating a new model set by modifying a design parameter of a prior model set and calculating a performance characteristic associated with the design parameter specified by the new model set.

25

48. The method of Claim 46 wherein said step of comparing includes:

- 40 -

assigning a cost to the calculated performance characteristic and using said cost to compare the calculated performance characteristic with the desired performance specification.

5      49. The method of Claim 39 wherein said step of constructing comprises: simulating the circuit element to determine the performance characteristics.

10      50. The method of Claim 49 wherein said circuit element is simulated using a SPICE simulator.

15      51. The method of Claim 47 wherein said step of creating a new model comprises:  
simulating the circuit element to determine the performance characteristics.

52. The method of Claim 51 wherein said circuit element is simulated using a SPICE simulator.

20      53. A method of using a database of circuit element design parameters for electronic circuit synthesis, comprising the steps of:

inputting a desired performance specification of a circuit element;  
retrieving from a database a learned model set which satisfies the desired performance specification if one exists, and if one does not exist the  
25      substeps of:

constructing a model set for the circuit element which links design parameters of said circuit element parameters to the performance characteristics for said circuit element;

- 41 -

and,

determining from the model set the design parameters which satisfy the desired performance specification.

- 5      54.    The method of Claim 53 wherein said step of inputting includes inputting a plurality of desired performance specification of the circuit element.
- 10      55.    The method of Claim 53 wherein said step of inputting includes specifying a value limit for the desired performance specification.
- 15      56.    The method of Claim 53 wherein said step of inputting includes specifying a target value for the desired performance specification.
- 15      57.    The method of Claim 53 wherein said step of inputting includes inputting a synthesis plan wherein said synthesis plan comprises a set of desired circuit element performance specification together with target values and value limits for each of said performance specification.
- 20      58.    The method of Claim 53 wherein said step of constructing includes: creating a model set table entry in the database wherein the table entry comprises a design parameter value and an associated performance characteristic value.
- 25      59.    The method of Claim 53 wherein said substep of constructing further comprises:  
         calculating a corresponding performance characteristic associated with a design parameter specified by the model set; and,

- 42 -

comparing the calculated performance characteristic with the desired performance specification.

5 60. The method of Claim 59 wherein said step of determining further comprises:

creating a new model by modifying a design parameter of a learned model and calculating a corresponding performance characteristic associated with a design parameter specified by the new model.

10 61. The method of Claim 59 wherein said step of comparing includes:  
assigning a cost to the calculated performance characteristic and  
using said cost to compare the calculated performance characteristic with  
the desired performance specification.

15 62. The method of Claim 59 wherein said step of calculating comprises:  
simulating the circuit element to determine the performance  
characteristics.

20 63. The method of Claim 62 wherein said circuit element is simulated  
using a SPICE simulator.

64. A learning circuit element design system for use in electronic circuit  
synthesis comprising:

25 an input device to allow input of a set of desired performance  
specification;

an evaluator for determining circuit element models which satisfies  
the set of desired performance specification; and,

- 43 -

a storage for storing models for use by the evaluator in subsequent evaluations, wherein each of said models includes a set of circuit element design parameters and associated circuit element performance characteristics.

5

65. The system of Claim 64 wherein the set of desired performance specification include targets and limits associated with each specification.

66. The system of Claim 64 wherein the evaluator comprises:

10 a starting point determiner, which determines an initial circuit element model by reference to a database of stored models;

a cost determiner, which calculates the cost of the circuit element model compared to the desired performance specification; and,

15 a model creator which creates circuit element models if none are available within the determined cost.

67. The system of Claim 64 further comprising:

20 a database updater, which optionally updates the entries of the database with learned entries that match the circuit element designs created by the evaluator.

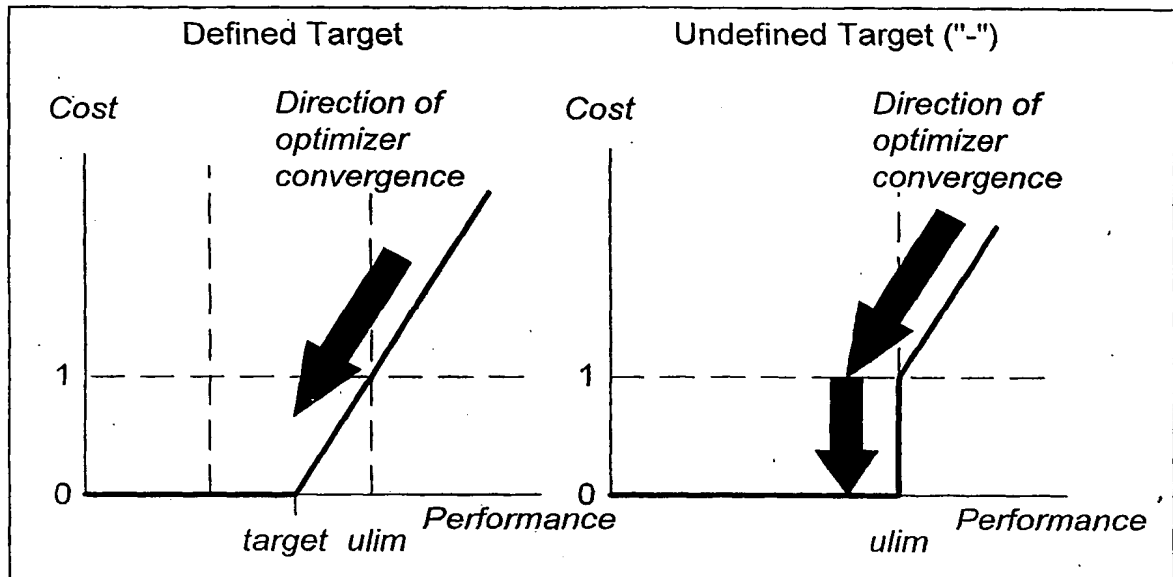
68. The system of Claim 66 wherein the model creator includes a simulator which use the circuit element design parameters to determine the performance characteristics of the circuit element.

25

69. The system of Claim 68 wherein the simulator is a SPICE simulator.

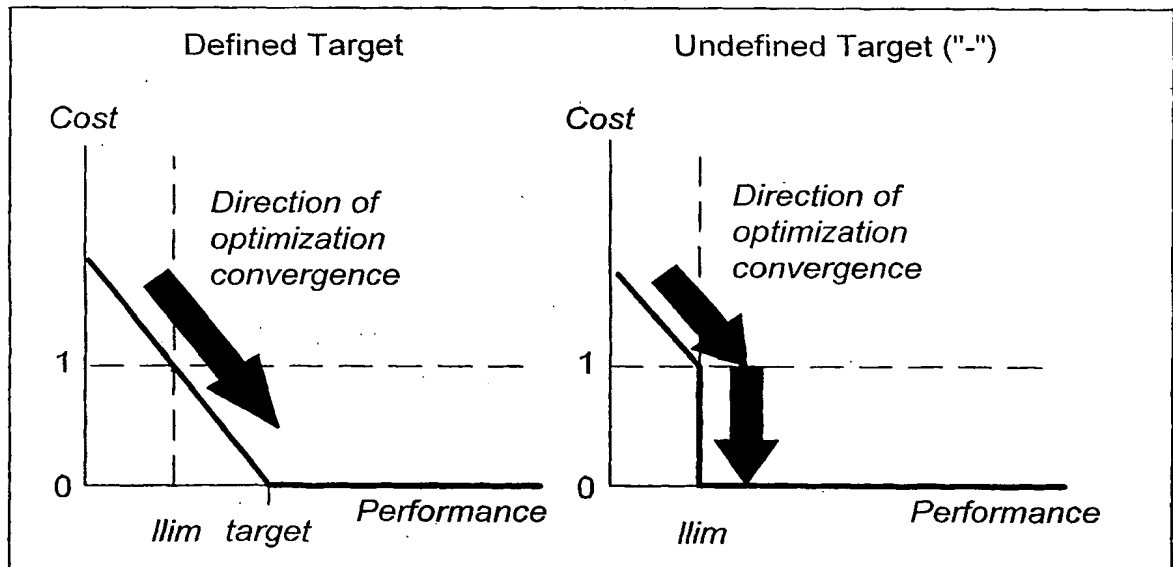
- 44 -

70. The system of Claim 67 wherein the option of causing the database updater to update the entries of the database with learned entries may be specified by the user of the system.
- 5      71. The method of Claim 53 further comprising the step of:  
storing the model set in a database as a learned model for use in subsequent circuit element design parameter determination.



\$ Cost for Minimize Optimizations (Defined or "-" Target)

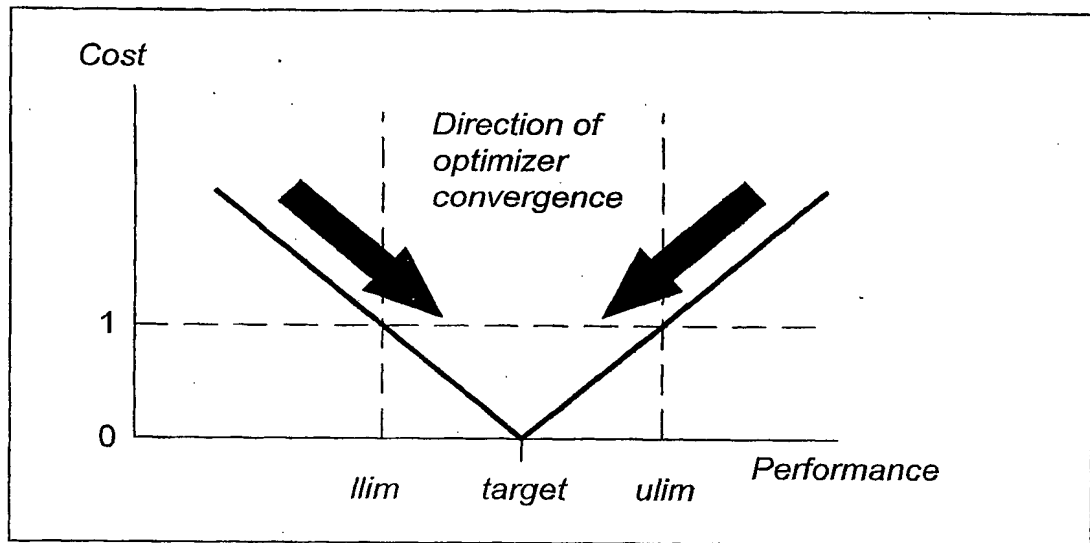
FIG. - 1



\$ Cost for Maximize Optimizations (Defined or "-" Target)

FIG. - 2

2/5



Cost Function for Goal Type Optimization

FIG. - 3

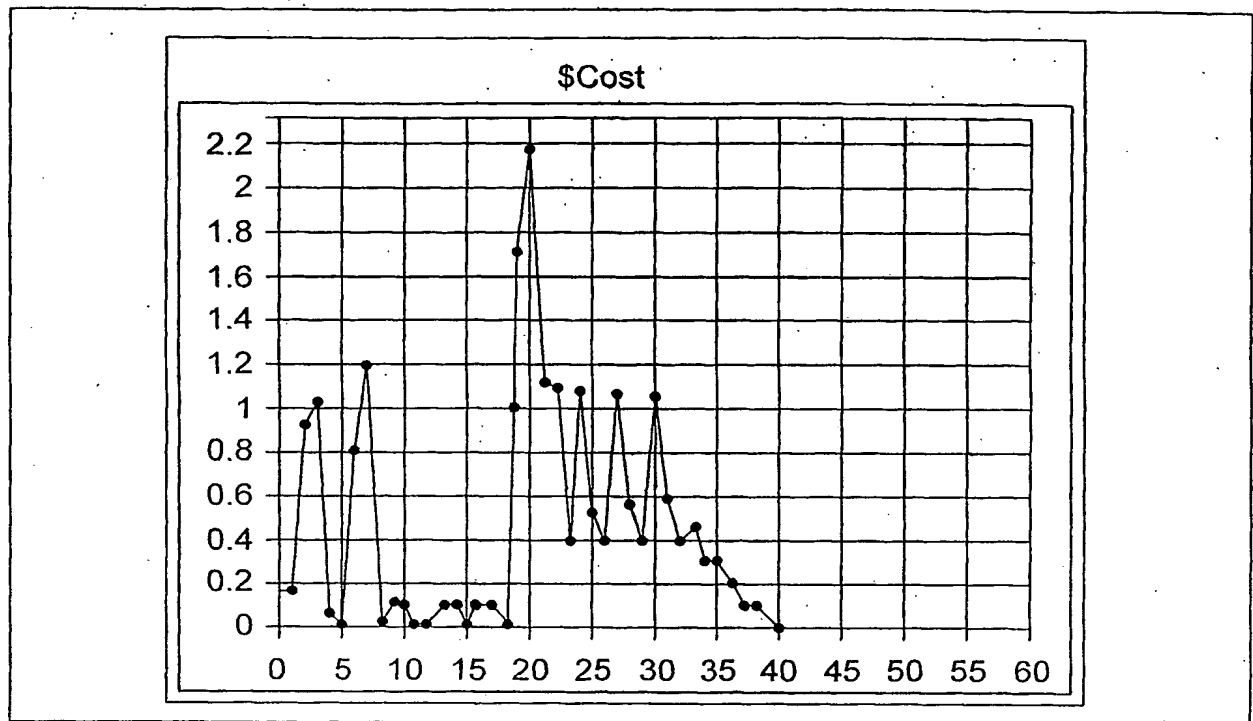
Eval count	Minimum cost
1	1.97822
2	1.97822
3	1.61137
4	1.34617
5	1.34617
6	0.81104
7	0.81104
8	0.81104
9	0.81104
10	0.81104
11	0.81104
12	0.81104
13	0.81104
14	0.81104
15	0.81104
16	0.81104
17	0.81104
18	0.75361
19	0.7335
20	0.72883

Iterations 1-2, 4-5, and 6-17 Show Lowest Cost while Exploring

FIG. - 4



3/5



Cost Graph Produced in the MSS Graphical Display

FIG. - 5

(wpdiff, wndiff, wpo, wno, rise, fall)

1e-06, 6.8e-06, 6.9e-06, 2e-05, 6.56271e-10, 5.21726e-10,  
 1e-06, 6.8e-06, 6.9e-06, 1.93e-05, 6.53844e-10, 5.22327e-10,  
 1e-06, 6.8e-06, 1.43e-05, 1.93e-05, 7.02569e-10, 6.27855e-10,  
 1e-06, 6.8e-06, 1.06e-05, 1.93e-05, 6.38513e-10, 5.46723e-10,  
 1e-06, 8.8e-06, 1.06e-05, 1.93e-05, 6.42615e-10, 5.43541e-10,  
 1.6e-06, 7.8e-06, 1.06e-05, 1.93e-05, 6.91275e-10, 5.6497e-10,  
 1e-06, 7.8e-06, 1.06e-05, 1.93e-05, 6.39836e-10, 5.45845e-10,  
 1.6e-06, 8.8e-06, 1.43e-05, 1.74e-05, 6.72262e-10, 5.81809e-10,  
 1.6e-06, 8.8e-06, 1.43e-05, 1.55e-05, 6.71928e-10, 5.9119e-10,  
 1.6e-06, 8.8e-06, 2.17e-05, 1.55e-05, 7.59561e-10, 7.20054e-10,  
 1.6e-06, 8.8e-06, 1.8e-05, 1.55e-05, 7.00919e-10, 6.45454e-10,  
 1.6e-06, 1.08e-05, 1.8e-05, 1.55e-05, 7.02305e-10, 6.46667e-10,

Contents of a Sample lookup db File

FIG. - 6

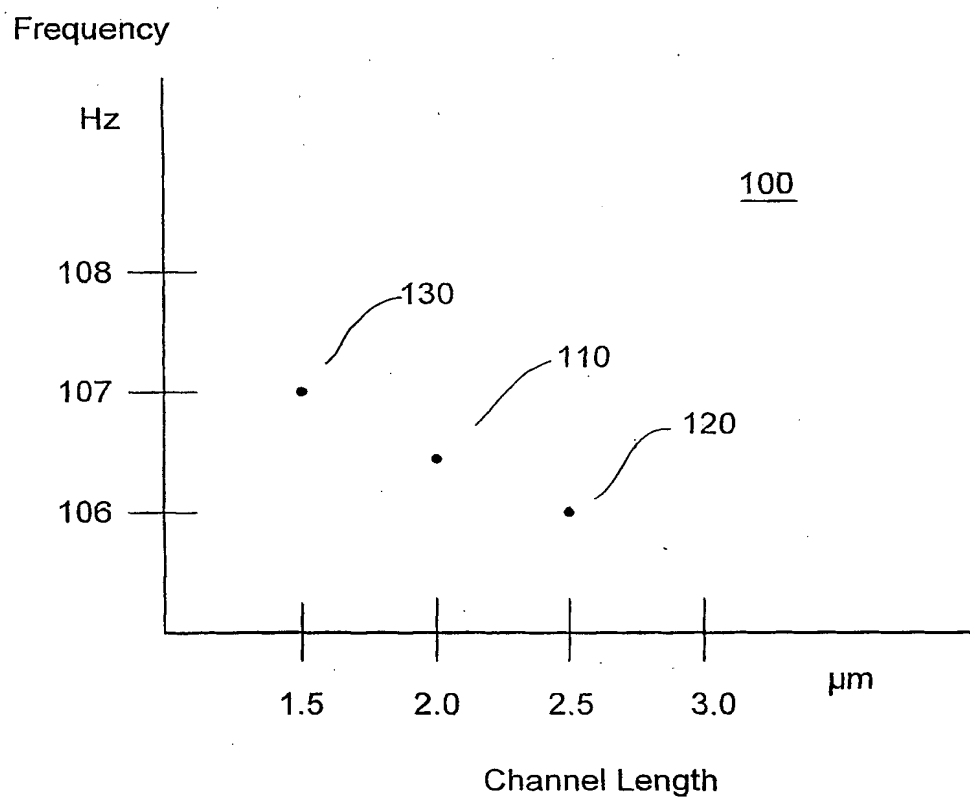


FIG. - 7

5/5

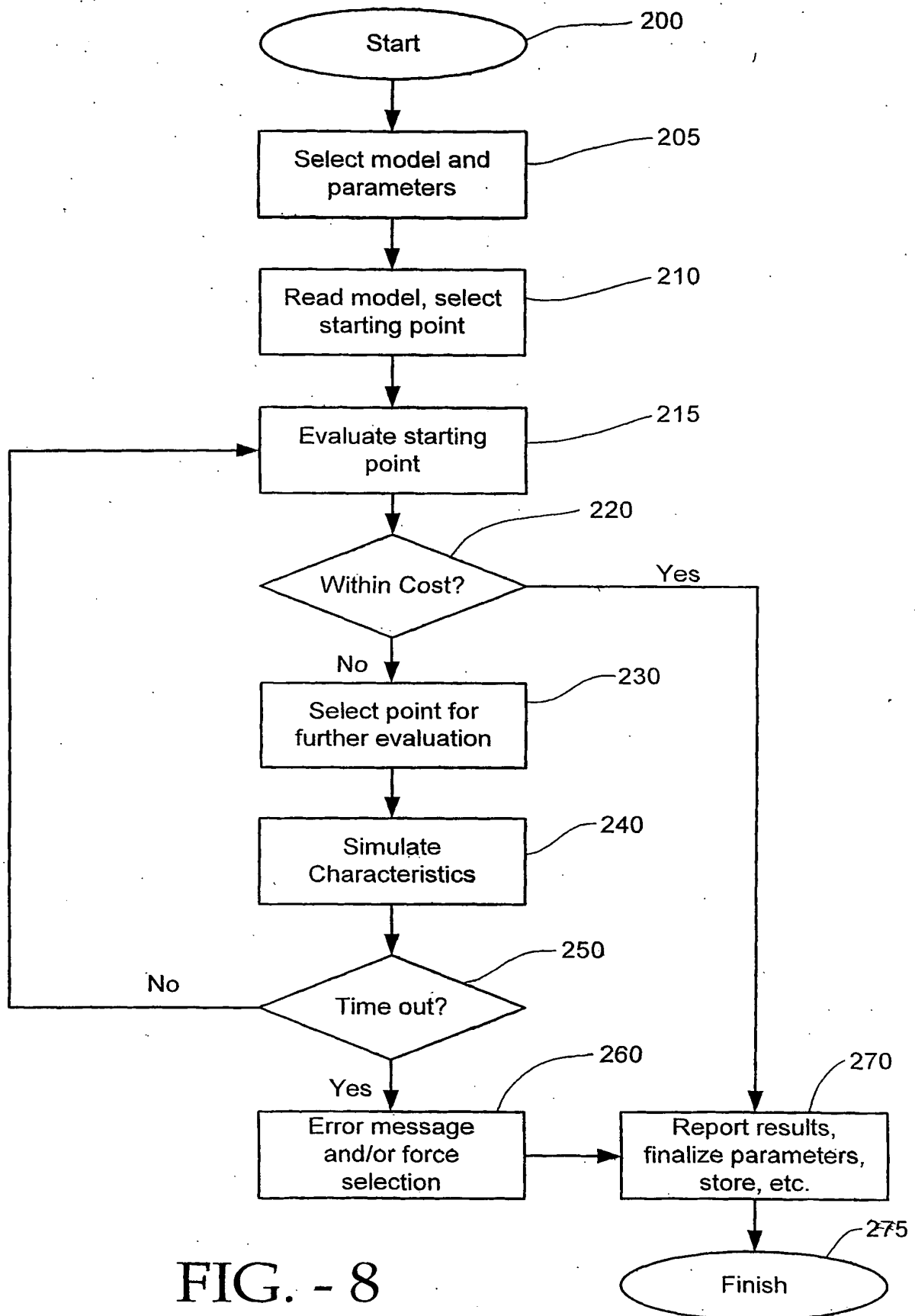


FIG. - 8

# INTERNATIONAL SEARCH REPORT

International application No.

PCT/US00/30980

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G05B 13/02

US CL : 700/29, 31, 32, 33, 45, 121; 716/2

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 700/29, 31, 32, 33, 45, 121; 716/2

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)  
Please See Continuation Sheet

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,898,595 A (BAIR ET AL.) 27 April 1999(27.04.1999), whole document	1-4 and 33-36
---		
Y		5-8, 37-38, 49-52 and 60-71
Y	US 5,563,928 A (ROSTOKER ET AL.) 08 October 1996 (08.10.1996), abstract and col. 1 line 17 to col. 6 line 2.	5-8 and 37-38
X	US 5,764,532 A (PATEL) 09 June 1998 (09.06.1998), whole document	9-16 and 24-28 and 64-70
---		
Y		17-23
Y	US 5,539,652 (TEGETHOFF) 23 July 1996 (23.07.1996), abstract, col. 1 line 7 to col. 8 line 67 and figs. 1-14.	17-23
X	US 5,878,383 A (CARTER) 02 March 1999 (02.03.1999), whole document	29-32
X	US 5,689,432 A (BLAAUW ET AL.) 18 November 1997 (18.11.1997), whole document	39-48 and 53-59
---		
Y		49-52 and 60-71

☐

Further documents are listed in the continuation of Box C.

☐

See patent family annex.

\* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T"

later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X"

document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y"

document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&"

document member of the same patent family

Date of the actual completion of the international search

08 January 2001 (08.01.2001)

Date of mailing of the international search report

07 MAR 2001

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703)305-3230

Authorized officer

William Grant

Telephone No. (703)305-9600

*Peggy Hand*

**Continuation of B. FIELDS SEARCHED Item 3:** East(performance characteristics, parameters, optimization, circuit, database, optimizer, input, output computer, storage, evaluate, model)

**THIS PAGE BLANK (USPTO)**